Quiz 4

Name: ______________________________

(2 + 8 = 10 points) Maximum weight independent set on path graphs

A path graph G is simply an undirected graph consisting of the vertex set $V = \{v_1, v_2, \ldots, v_n\}$ and the edge set

$$E = \{(v_i, v_{i+1}) : 1 \leq i \leq n - 1\}.$$

An *independent set* of G is a subset I of V such that there is no edge in E between any pair of vertices of I. We also associate a positive weight w_i with vertex v_i. The weight of any subset S of vertices is simply the sum of the weights of the vertices in S.

For example, consider a path graph G with five vertices v_1, v_2, v_3, v_4, v_5, with weights 1, 9, 6, 3, and 7, respectively and four edges $(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_5)$. The set $\{v_1, v_4\}$ is an independent set with weight 4, while the set $\{v_1, v_3, v_4\}$ is not an independent set.

This exercise concerns an algorithm for finding a maximum-weight independent set of a given path graph G.

(a) Consider the following greedy algorithm.

1. Initialize the independent set I to be empty.
2. While some node remains in G, add a node v in G of highest weight to I, and remove v and its neighbors from G.
3. Return I.

Give a counterexample that shows that the above greedy algorithm does not return a maximum-weight independent set.

Answer: An counterexample is the following: Consider a path graph G with three vertices v_1, v_2, v_3, with weights 3, 5, 4 respectively, and two edges (v_1, v_2), (v_2, v_3).

According to the greedy algorithm, it will first pick up v_2 into I, and remove v_1 and v_3 which are connected to v_2. Then return I, and its weight is 5.

Obviously this I is not the maximum-weight independent set, since we have v_1, v_3 whose weight is 7, bigger than the I returned by the greedy algorithm.

(b) Give a polynomial-time dynamic programming algorithm to solve the problem. It is sufficient to give a recurrence, and describe your algorithm briefly in words. State the running time of your algorithm.

Answer: We denote $m(k)$ as the subproblem of the original problem, which represents finding the maximum-weight independent set from the first k vertices and the subgraph consisted by them.

Initially we have:

$$\begin{aligned} m(0) &= 0 \\ m(1) &= \max\{w_1, 0\} \end{aligned}$$

Note, we need to consider if w_1 is positive.

Recursively, when we consider $m(k)$, we have two possibilities:

- If we choose v_k, then we have $m(k) = m(k-2) + w_k$(if w_k is negative, we have $m(k) = m(k-2)$)
- If we don't choose v_k, then we have $m(k) = m(k-1)$

So we have the following equations:

$$m(k) = \max\{m(k-1), m(k-2) + w_k\}$$

Thus we need only one-pass to traverse all the $m(k)$, and got the solution in $O(n)$ time.